G.C.E. O/L Grade 10 ICT

Lesson 3: Data Representation in Computer Systems

- 3.1 Binary – Representing Data with Two States
- 3.2 Decimal to Binary, Octal and Hexadecimal
- 3.3 Binary ↔ Decimal, Octal and Hexadecimal Conversions
- 3.4 Data Storage Units – Bit, Byte, KB, MB, GB, TB
- 3.5 Coding Systems – BCD, EBCDIC, ASCII and Unicode

© 2026 schoolict.net (Ruwan Mangala Suraweera)
All rights reserved.

G.C.E. O/L Grade 10 ICT · Lesson 3

3.1 Binary – Representing Data with Two States

Inside a computer, everything – numbers, letters, photos, songs and videos – is stored using only two symbols: **1 and 0**. In this note you will learn why computers use two states, what the **binary number system** is, and how to write a positive decimal number in binary.

1. Data and two states

A computer is made of millions (even billions) of tiny electronic switches called **transistors**. Each switch can be in only **two states**: ON or OFF. We write ON as **1** and OFF as **0**. Because there are only two states, it is easy and reliable to store and send them.

State 1	State 0	Where you see it
ON	OFF	Switch / transistor
TRUE / YES	FALSE / NO	Logic, decisions
High voltage	Low voltage	Circuits, wires
Magnetised one way	Magnetised other way	Hard disk
Light reflected / pit	No change	CD / DVD
Charge stored	No charge	SSD, USB flash, memory card

💡 **Digital** data = data stored as separate values 0 and 1. A single 0 or 1 is called a **bit** (binary digit).

2. Why do computers use binary?

- Electronic switches have only two states – ON/OFF fits perfectly.
- Two states are easy to tell apart, so there are fewer errors (even with electrical noise).
- Circuits are simpler, cheaper and faster to build.
- All kinds of data (text, images, sound, video) can be turned into patterns of 1s and 0s.

3. The binary number system

We normally use the **decimal** system (base 10) with ten digits 0–9. The **binary** system has **base 2** and only two digits: **0 and 1**. Each place is worth **2 times** the place on its right: 1, 2, 4, 8, 16, 32, 64, 128 ...

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

To show which system a number belongs to, write the base as a small number: 1101_2 (binary), 13_{10} (decimal). Without a base, 10 could mean ten or two!

4. Writing a positive decimal integer in binary

Method 1 – place values: find the biggest place value that fits, put 1 there, subtract, and continue. Example: 13 = 8 + 4 + 1.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8	4	2	1
1	1	0	1

💡 So 13_{10} = **1101_2**. Check: 8 + 4 + 0 + 1 = 13.

Method 2 – repeated division by 2: divide by 2, write the remainder, keep dividing the quotient until it becomes 0. Read the remainders from **bottom to top**. Example: 25_{10}.

Division	Quotient	Remainder
25 ÷ 2	12	1
12 ÷ 2	6	0
6 ÷ 2	3	0
3 ÷ 2	1	1
1 ÷ 2	0	1

💡 Reading remainders bottom to top: 25_{10} = **11001_2**. (16 + 8 + 1 = 25)

5. How many values can n bits hold?

With **n bits** you can make **2^n** different patterns. 1 bit → 2, 2 bits → 4, 3 bits → 8, 8 bits → 256 (0 to 255). The largest number with n bits is **$2^n - 1$**.

Bits (n)	Patterns (2^n)	Range
1	2	0 – 1
2	4	0 – 3
3	8	0 – 7
4	16	0 – 15
8	256	0 – 255

NEW 2026 **2026 link:** a photo on your phone is millions of tiny dots (pixels), each stored as bits for red, green and blue. A 12-megapixel photo is about 36 million bytes before compression!

Key Words

Binary – Number system with base 2 using only digits 0 and 1.

Bit – A single binary digit, 0 or 1 – the smallest unit of data.

Two-state system – A system with only two conditions such as ON/OFF or TRUE/FALSE.

Transistor – A tiny electronic switch inside a chip that is ON (1) or OFF (0).

Decimal – Number system with base 10 using digits 0–9.

Base (radix) – The number of different digits a number system uses.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Place value – The value of a digit's position, e.g. 1, 2, 4, 8 in binary.

Repeated division – Dividing by the base again and again and reading the remainders upward.

Remainder – What is left after dividing; in base 2 it is 0 or 1.

Digital data – Data stored as separate (discrete) values such as 0 and 1.

MSB – Most Significant Bit – the left-most bit with the highest place value.

LSB – Least Significant Bit – the right-most bit with place value 1.

Sample Q&A

1. Why do computers use the binary system?

> **Answer:** Electronic switches (transistors) have only two states, ON and OFF, which map to 1 and 0. Two states are easy to detect reliably and circuits are simpler.

2. Give three pairs of two states that can represent 1 and 0.

> **Answer:** ON/OFF, TRUE/FALSE, high voltage/low voltage (also YES/NO, magnetised/not).

3. Convert 13_{10} to binary.

> **Answer:** 13 = 8 + 4 + 1 → 1101_2.

4. What is a bit?

> **Answer:** A binary digit – a single 0 or 1, the smallest unit of data.

5. How many different values can 8 bits represent?

> **Answer:** 2^8 = 256 values (0 to 255).

6. In repeated division, in which order are the remainders read?

> **Answer:** From the last remainder (bottom) to the first (top). The last remainder is the MSB.

MCQ

1. How many digits does the binary number system use?

A. 2
B. 8
C. 16
D. 10

2. Which pair best describes the two states used in a computer?

A. Red and blue
B. ON and OFF
C. Up, down and middle
D. 0 to 9

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. A single binary digit is called a:

A. Word
B. Nibble
C. Bit
D. Byte

4. What is 13_{10} in binary?

A. 1101
B. 1001
C. 1110
D. 1011

5. What is 25_{10} in binary?

A. 11001
B. 10101
C. 11010
D. 10011

6. What is the place value of the 5th bit from the right in binary?

A. 5
B. 32
C. 16
D. 10

7. How many different patterns can 4 bits make?

A. 15
B. 16
C. 8
D. 4

8. The largest number that can be stored in 8 bits is:

A. 256
B. 128
C. 8
D. 255

9. Which decimal number is the same as 1010_2?

A. 12
B. 5
C. 10
D. 1010

10. When converting by repeated division by 2, the answer is read:

A. Remainders from top to bottom
B. Only the last quotient
C. Quotients from top to bottom
D. Remainders from bottom to top

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

11. Why is binary reliable for electronic circuits?

A. Only two states are easy to tell apart
B. It is the base used by humans
C. It uses ten voltage levels
D. It needs no electricity

12. Which number CANNOT be a binary number?

A. 1001
B. 1021
C. 10
D. 1111

13. In the binary number 10110_2, the MSB is in the place worth:

A. 32
B. 1
C. 2
D. 16

14. How many bits are needed at minimum to write 6_{10} in binary?

A. 2
B. 3
C. 6
D. 4

Structured Questions

1. (i) What is meant by a two-state system? (1 mark)
(ii) Give two examples of two states used to represent 1 and 0. (2 marks) *(3 marks)*

> (i) A system that has only two possible conditions.
> (ii) ON/OFF, TRUE/FALSE, high/low voltage, magnetised/not magnetised (any two).

2. Give three reasons why computers use the binary number system. *(3 marks)*

> Transistors have only two states; two states are easy to detect with fewer errors; circuits are simpler/cheaper/faster; all data types can be coded as 0s and 1s.

3. Convert 45_{10} to binary using repeated division. Show your working. (3 marks) *(3 marks)*

> 45÷2=22 r1; 22÷2=11 r0; 11÷2=5 r1; 5÷2=2 r1; 2÷2=1 r0; 1÷2=0 r1. Read upward: $45_{10} = 101101_2$.

4. Write 100_{10} in binary using place values. (3 marks) *(3 marks)*

> 100 = 64 + 32 + 4 → 64 32 16 8 4 2 1 = 1 1 0 0 1 0 0 → 1100100_2.

5. (i) How many different values can be stored in 5 bits? (1 mark)
(ii) What is the largest decimal number that can be stored in 5 bits? (1 mark)
(iii) Write that number in binary. (1 mark) *(3 marks)*

> (i) $2^5 = 32$ (ii) 31 (iii) 11111_2

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. A smart bulb in a classroom is controlled by an app. (i) How can the bulb's ON/OFF state be stored as one bit? (1 mark) (ii) Eight bulbs are in a row. How many different ON/OFF patterns are possible? (2 marks) *(3 marks)*

> (i) ON = 1, OFF = 0. (ii) Each bulb is 1 bit, 8 bits → 2^8 = 256 patterns.

Summary

- Computers store all data using two states: 1 and 0.
- Two states: ON/OFF, TRUE/FALSE, high/low voltage.
- A bit is one binary digit; it is the smallest unit of data.
- Binary has base 2; place values 1, 2, 4, 8, 16, 32, 64, 128.
- Decimal → binary: place values or repeated division by 2.
- Read the remainders from bottom to top.
- n bits give 2^n patterns; the largest value is $2^n - 1$.
- Write the base as a subscript, e.g. $1101_2 = 13_{10}$.

MCQ answer key – 3.1

1. **A** – Binary has base 2: digits 0 and 1.
2. **B** – A switch is either ON (1) or OFF (0).
3. **C** – Bit = binary digit.
4. **A** – 13 = 8 + 4 + 1 = 1101_2.
5. **A** – 25 = 16 + 8 + 1 = 11001_2.
6. **C** – Place values 1, 2, 4, 8, 16 → 2^4 = 16.
7. **B** – 2^4 = 16.
8. **D** – $2^8 - 1$ = 255 (11111111_2).
9. **C** – 8 + 2 = 10.
10. **D** – The last remainder is the MSB, so read upward.
11. **A** – Two levels are easy to detect even with noise.
12. **B** – Binary uses only 0 and 1; '2' is not allowed.
13. **D** – 5 bits: 16, 8, 4, 2, 1 → MSB place = 16.
14. **B** – 6 = 110_2 → 3 bits.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

G.C.E. O/L Grade 10 ICT · Lesson 3

3.2 Decimal to Binary, Octal and Hexadecimal

Computers use binary, but long strings of 1s and 0s are hard for people to read. So programmers also use **octal (base 8)** and **hexadecimal (base 16)**. In this note you will learn the four number systems, their **place values**, and how to convert a decimal number to any of them using **repeated division**.

1. Four number systems

System	Base	Digits used	Example
Binary	2	0, 1	1011_2
Octal	8	0 – 7	157_8
Decimal	10	0 – 9	249_{10}
Hexadecimal	16	0 – 9, A – F	$3F_{16}$

The **base** tells you how many digits the system has. The biggest digit is always **base − 1** (1 in binary, 7 in octal, 9 in decimal, F in hex).

2. Hexadecimal digits A–F

Hex needs 16 single digits, so after 9 it uses letters:

Hex	A	B	C	D	E	F
Decimal	10	11	12	13	14	15

3. Place values

In every system the right-most place is worth 1 (base^0). Each place to the left is worth **base times** more.

Base	4th place	3rd place	2nd place	1st place
2	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
8	$8^3 = 512$	$8^2 = 64$	$8^1 = 8$	$8^0 = 1$
10	$10^3 = 1000$	$10^2 = 100$	$10^1 = 10$	$10^0 = 1$
16	$16^3 = 4096$	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$

> 💡 Example: in 234_8 the 2 is worth 2 × 64, the 3 is worth 3 × 8 and the 4 is worth 4 × 1.

4. Repeated division method

- Divide the decimal number by the **new base** (2, 8 or 16).
- Write down the **remainder**.
- Divide the **quotient** again; repeat until the quotient is 0.
- Read the remainders from **bottom to top**. In hex, change 10–15 to A–F.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Worked example 1: 47_{10} → binary

Division	Quotient	Remainder
47 ÷ 2	23	1
23 ÷ 2	11	1
11 ÷ 2	5	1
5 ÷ 2	2	1
2 ÷ 2	1	0
1 ÷ 2	0	1

💡 47_{10} = **101111_2**

Worked example 2: 156_{10} → octal

Division	Quotient	Remainder
156 ÷ 8	19	4
19 ÷ 8	2	3
2 ÷ 8	0	2

💡 156_{10} = **234_8**. Check: 2×64 + 3×8 + 4×1 = 128 + 24 + 4 = 156.

Digit	2	3	4
Place value	$8^2 = 64$	$8^1 = 8$	$8^0 = 1$
Digit × place value	2 × 64 = 128	3 × 8 = 24	4 × 1 = 4

Worked example 3: 500_{10} → hexadecimal

Division	Quotient	Remainder
500 ÷ 16	31	4
31 ÷ 16	1	15 = F
1 ÷ 16	0	1

💡 Reading bottom to top: 1, F, 4 → 500_{10} = **$1F4_{16}$**. Check: 1×256 + 15×16 + 4 = 500.

💡 **Exam trick:** always check your answer by multiplying each digit by its place value and adding. If you get the original number, you are right!

5. Comparing the same number

Decimal	Binary	Octal	Hex
8	1000	10	8
10	1010	12	A

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

15	1111	17	F
16	10000	20	10
64	1000000	100	40
100	1100100	144	64
255	11111111	377	FF

NEW 2026 **Where you see hex today:** web colours (#FF0000 is red), MAC addresses on Wi-Fi routers (e.g. 3C:5A:B4:...), IPv6 addresses and error codes on your phone or PC.

Key Words

Octal – Number system with base 8 using digits 0–7.

Hexadecimal – Number system with base 16 using 0–9 and A–F.

Base – The number of digits in a number system (2, 8, 10, 16).

Place value – Value of a position: a power of the base (base^0, base^1 ...).

Quotient – The whole-number answer of a division.

Remainder – The amount left over after a division; always smaller than the base.

Repeated division – Method of dividing by the new base until the quotient is 0.

Subscript – Small number after a value showing its base, e.g. 75_8.

Hex digit A–F – Letters that stand for 10–15 in hexadecimal.

Power of the base – The base multiplied by itself, e.g. $16^2 = 256$.

Decimal – The base-10 system people use every day.

Colour code – A 6-digit hex value for a colour on a screen, e.g. #00FF00.

Sample Q&A

1. What is the base of the octal number system and which digits does it use?

Answer: Base 8; digits 0, 1, 2, 3, 4, 5, 6, 7.

2. What do the letters A to F stand for in hexadecimal?

Answer: A=10, B=11, C=12, D=13, E=14, F=15.

3. Convert 100_{10} to octal.

Answer: 100÷8=12 r4; 12÷8=1 r4; 1÷8=0 r1 → 144_8.

4. Convert 250_{10} to hexadecimal.

Answer: 250÷16=15 r10(A); 15÷16=0 r15(F) → FA_{16}.

5. What are the first four place values in hexadecimal?

Answer: 1, 16, 256, 4096 (16^0 to 16^3).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. Why do programmers use hexadecimal?

> **Answer:** It is much shorter than binary and easy to convert (one hex digit = 4 bits), e.g. colour codes and memory addresses.

MCQ

1. What is the base of the hexadecimal system?

A. 6
B. 16
C. 8
D. 10

2. Which digit is NOT used in the octal system?

A. 0
B. 7
C. 5
D. 8

3. In hexadecimal, the letter D stands for:

A. 13
B. 11
C. 14
D. 12

4. What is 156_{10} in octal?

A. 243
B. 234
C. 324
D. 432

5. What is 250_{10} in hexadecimal?

A. FA
B. AF
C. F10
D. FB

6. What is 47_{10} in binary?

A. 110111
B. 101101
C. 101111
D. 111101

7. The place value of the 3rd digit from the right in octal is:

A. 64
B. 512
C. 16
D. 24

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. The place value of the 3rd digit from the right in hexadecimal is:

A. 32
B. 48
C. 4096
D. 256

9. What is 255_{10} in hexadecimal?

A. 100
B. FF
C. F5
D. EF

10. What is 64_{10} in octal?

A. 100
B. 010
C. 80
D. 64

11. What is 200_{10} in hexadecimal?

A. 8C
B. 128
C. C8
D. C2

12. When converting to hexadecimal, a remainder of 11 is written as:

A. 11
B. C
C. B
D. A

13. Which of these is a valid hexadecimal number?

A. 1H
B. 2E9
C. G5
D. Z0

14. 16_{10} written in hexadecimal is:

A. 16
B. F
C. G
D. 10

15. A web colour #FF0000 uses which number system?

A. Octal
B. Binary
C. Hexadecimal
D. Roman

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Structured Questions

1. Copy and complete the table:
System | Base | Digits
Binary | ? | ?
Octal | ? | ?
Hexadecimal | ? | ? (4 marks) *(4 marks)*

> Binary – 2 – 0, 1
> Octal – 8 – 0 to 7
> Hexadecimal – 16 – 0 to 9 and A to F

2. Convert 83_{10} to octal using repeated division. Show working. (3 marks) *(3 marks)*

> 83÷8=10 r3; 10÷8=1 r2; 1÷8=0 r1 → 123_8. Check: 64+16+3=83.

3. Convert 175_{10} to hexadecimal. Show working. (3 marks) *(3 marks)*

> 175÷16=10 r15(F); 10÷16=0 r10(A) → AF_{16}. Check: 10×16+15=175.

4. Convert 29_{10} to (i) binary (ii) octal (iii) hexadecimal. (3 marks) *(3 marks)*

> (i) 11101_2 (ii) 35_8 (iii) $1D_{16}$

5. (i) Write the place values of the first four positions in octal. (2 marks)
(ii) What is the largest single digit in base 8 and base 16? (2 marks) *(4 marks)*

> (i) 1, 8, 64, 512 (ii) base 8 → 7; base 16 → F (15)

6. A web designer wants the colour with red = 200, green = 30, blue = 255. Colour codes are written as #RRGGBB in hex. (i) Convert 200, 30 and 255 to hex. (3 marks) (ii) Write the colour code. (1 mark) *(4 marks)*

> (i) 200 = C8, 30 = 1E, 255 = FF (ii) #C81EFF

Summary

- Four number systems: binary (2), octal (8), decimal (10), hexadecimal (16).
- The largest digit in a system is base − 1.
- Hex uses A–F for 10–15.
- Place values are powers of the base: 1, base, base^2, base^3 …
- Octal place values: 1, 8, 64, 512. Hex: 1, 16, 256, 4096.
- Decimal → any base: divide by the base repeatedly, read remainders upward.
- Always check by multiplying digits by place values and adding.
- Hex is used for colour codes, MAC and IPv6 addresses.

MCQ answer key – 3.2

1. **B** – Hexa (6) + decimal (10) = 16.
2. **D** – Octal digits are 0–7.
3. **A** – A10 B11 C12 D13.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

4. **B** – Remainders 4, 3, 2 read upward → 234_8.
5. **A** – 250 = 15×16 + 10 → FA.
6. **C** – 32 + 8 + 4 + 2 + 1 = 47.
7. **A** – 8^0=1, 8^1=8, 8^2=64.
8. **D** – 16^2 = 256.
9. **B** – 255 = 15×16 + 15 → FF.
10. **A** – 64 = 1×8^2 → 100_8.
11. **C** – 200 = 12×16 + 8 → C8.
12. **C** – B = 11.
13. **B** – Hex digits are 0–9 and A–F only.
14. **D** – 16 = 1×16 + 0 → 10_{16}.
15. **C** – Colour codes are 6 hex digits (RRGGBB).

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

G.C.E. O/L Grade 10 ICT · Lesson 3

3.3 Binary ↔ Decimal, Octal and Hexadecimal Conversions

Now we go the other way. You will convert **binary to decimal** using place values, and convert **binary ↔ octal** and **binary ↔ hexadecimal** quickly by **grouping bits in 3s and 4s**. Finally you will convert **octal ↔ hexadecimal** through binary. Only positive whole numbers are used.

1. Binary → decimal (place value method)

Write the place values (1, 2, 4, 8 ...) under each bit from the right. Add the place values where the bit is **1**.

Worked example 1: 101101_2 → decimal

Digit	1	0	1	1	0	1
Place value	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Digit × place value	1 × 32 = 32	0 × 16 = 0	1 × 8 = 8	1 × 4 = 4	0 × 2 = 0	1 × 1 = 1

💡 32 + 0 + 8 + 4 + 0 + 1 = **45_{10}**

The same method works for octal and hex → decimal: multiply each digit by its place value and add.

Worked example 2: $3A7_{16}$ → decimal

Digit	3	A	7
Place value	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
Digit × place value	3 × 256 = 768	10 × 16 = 160	7 × 1 = 7

💡 768 + 160 + 7 = **935_{10}**

2. Binary ↔ octal (groups of 3)

Because $8 = 2^3$, **one octal digit = exactly 3 bits**. Group the bits in threes **from the right**; add 0s on the left if needed. Change each group to one digit.

Octal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

Worked example 3: 1101011_2 → octal

Group	001	101	011
Value	1	5	3

💡 1101011_2 = **153_8** (two 0s were added on the left to make 001).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Worked example 4: 572_8 → binary

Digit	5	7	2
Bits	101	111	010

💡 572_8 = **101111010_2**

3. Binary ↔ hexadecimal (groups of 4)

Because 16 = 2^4, **one hex digit = exactly 4 bits** (a nibble). Group from the right in fours.

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Worked example 5: 1011111010_2 → hexadecimal

Group	0010	1111	1010
Value	2	F	A

💡 1011111010_2 = **$2FA_{16}$**

Worked example 6: $9C_{16}$ → binary

Digit	9	C
Bits	1001	1100

💡 $9C_{16}$ = **10011100_2**

4. Octal ↔ hexadecimal (via binary)

There is no direct grouping between 8 and 16. Go through binary: **octal → binary (3 bits each) → regroup in 4s → hex**, and the reverse for hex → octal.

Worked example 7: 357_8 → hexadecimal

Digit	3	5	7
Bits	011	101	111

Group	1110	1111
Value	E	F

💡 357_8 = 11101111_2 = **EF_{16}** (check: both equal 239_{10}).

Worked example 8: $2B_{16}$ → octal

Digit	2	B
Bits	0010	1011

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Group	101	011
Value	5	3

💡 $2B_{16} = 101011_2 = \mathbf{53_8}$

💡 **Common mistakes:** grouping from the LEFT instead of the right; forgetting leading 0s; writing 10 instead of A in hex.

Key Words

Grouping – Splitting a binary number into groups of 3 or 4 bits from the right.

Nibble – A group of 4 bits = one hexadecimal digit.

Triad – A group of 3 bits = one octal digit.

Leading zero – A 0 added on the left to complete a group; it does not change the value.

Place value method – Multiply each digit by its place value and add to get decimal.

Binary to decimal – Add the place values where the bit is 1.

Binary to octal – Group bits in 3s from the right and convert each group.

Binary to hexadecimal – Group bits in 4s from the right and convert each group.

Octal to binary – Write each octal digit as 3 bits.

Hexadecimal to binary – Write each hex digit as 4 bits.

Octal to hexadecimal – Convert octal to binary, then regroup in 4s.

Positive integer – A whole number greater than zero (no fractions, no minus sign).

Sample Q&A

1. Convert 11010_2 to decimal.

Answer: $16 + 8 + 2 = 26_{10}$.

2. Why do we group bits in 3s for octal?

Answer: Because $8 = 2^3$, so each octal digit is exactly 3 bits.

3. Convert 11110110_2 to hexadecimal.

Answer: 1111 | 0110 → F | 6 → $F6_{16}$.

4. Convert 47_8 to binary.

Answer: 4 → 100, 7 → 111 → 100111_2.

5. From which side do you start grouping bits?

Answer: From the right (the LSB). Add leading 0s on the left if the last group is short.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. How do you convert octal to hexadecimal?

> **Answer:** Change each octal digit to 3 bits, then regroup the bits in 4s from the right and change each group to a hex digit.

MCQ

1. What is 101101_2 in decimal?

A. 45
B. 90
C. 54
D. 43

2. What is 1111_2 in decimal?

A. 16
B. 1111
C. 15
D. 4

3. What is 1100100_2 in decimal?

A. 100
B. 200
C. 98
D. 50

4. What is 1101011_2 in octal?

A. 513
B. 153
C. 651
D. 326

5. What is 1011111010_2 in hexadecimal?

A. 2EA
B. 2FA0
C. 2FA
D. BE2

6. What is 572_8 in binary?

A. 101011010
B. 111101010
C. 101111001
D. 101111010

7. What is $9C_{16}$ in binary?

A. 10010011
B. 11001001
C. 10011100
D. 10011010

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. When converting binary to hexadecimal, bits are grouped in:

A. Eights from the left
B. Fours from the left
C. Threes from the right
D. Fours from the right

9. What is 357_8 in hexadecimal?

A. 357
B. EF
C. FE
D. DF

10. What is $2B_{16}$ in octal?

A. 53
B. 52
C. 35
D. 43

11. What is $3A7_{16}$ in decimal?

A. 937
B. 935
C. 3107
D. 967

12. What is 11110110_2 in hexadecimal?

A. 6F
B. F5
C. F6
D. E6

13. One hexadecimal digit is equal to how many bits?

A. 4
B. 8
C. 16
D. 3

14. What is 75_8 in decimal?

A. 117
B. 57
C. 75
D. 61

15. Which binary number equals $7F_{16}$?

A. 11110111
B. 1111111
C. 1111111111
D. 01110000

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Structured Questions

1. Convert 10110011_2 to (i) decimal (ii) octal (iii) hexadecimal. Show working. (6 marks) *(6 marks)*

> (i) 128+32+16+2+1 = 179_{10}
> (ii) 010 110 011 → 263_8
> (iii) 1011 0011 → $B3_{16}$

2. Convert 645_8 to binary and then to hexadecimal. (4 marks) *(4 marks)*

> 6→110, 4→100, 5→101 → 110100101_2. Regroup: 0001 1010 0101 → $1A5_{16}$.

3. Convert $A5_{16}$ to (i) binary (ii) octal (iii) decimal. (3 marks) *(3 marks)*

> (i) 1010 0101 → 10100101_2 (ii) 010 100 101 → 245_8 (iii) 10×16+5 = 165_{10}

4. A student converted 1101101_2 to hex as 1101 101 → D5. (i) What mistake was made? (1 mark) (ii) Give the correct answer. (2 marks) *(3 marks)*

> (i) Grouped from the left instead of the right. (ii) 0110 1101 → $6D_{16}$.

5. Arrange in ascending order after converting to decimal: 1110_2, 17_8, 10_{16}. (3 marks) *(3 marks)*

> 1110_2 = 14, 17_8 = 15, 10_{16} = 16 → 1110_2, 17_8, 10_{16}.

6. Convert $1F_{16}$ to octal using binary. Show both steps. (3 marks) *(3 marks)*

> 1→0001, F→1111 → 11111_2. Regroup in 3s: 011 111 → 37_8.

Summary

- Binary → decimal: add place values where the bit is 1.
- Octal/hex → decimal: digit × place value, then add.
- 1 octal digit = 3 bits ($8 = 2^3$).
- 1 hex digit = 4 bits ($16 = 2^4$), called a nibble.
- Always group from the right; add leading 0s on the left.
- Octal/hex → binary: write each digit as 3 or 4 bits.
- Octal ↔ hex: go through binary and regroup.
- Check answers by converting both to decimal.

MCQ answer key – 3.3

1. **A** – 32+8+4+1 = 45.
2. **C** – 8+4+2+1 = 15.
3. **A** – 64+32+4 = 100.
4. **B** – 001 101 011 → 1 5 3.
5. **C** – 0010 1111 1010 → 2 F A.
6. **D** – 5→101, 7→111, 2→010.
7. **C** – 9→1001, C→1100.
8. **D** – $16 = 2^4$; start at the LSB.
9. **B** – $357_8 = 11101111_2 = EF_{16}$.
10. **A** – $2B = 101011_2 = 101\ 011 = 53_8$.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

11. **B** – 3×256 + 10×16 + 7 = 935.
12. **C** – 1111 0110 → F6.
13. **A** – 16 = 2^4 → 4 bits.
14. **D** – 7×8 + 5 = 61.
15. **B** – 7→0111, F→1111 → 1111111.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

G.C.E. O/L Grade 10 ICT · Lesson 3

3.4 Data Storage Units – Bit, Byte, KB, MB, GB, TB

How big is a photo, a song or your phone's storage? In this note you will learn the **units of storage** – bit, byte, kilobyte, megabyte, gigabyte and terabyte – how to **convert between them** using **1 KB = 1024 bytes**, and how to **compare the capacity** of cache, RAM, ROM, SSD/hard disk, CD/DVD, USB drives, memory cards and cloud storage.

1. Bit and byte

A **bit** (0 or 1) is the smallest unit. **8 bits = 1 byte**. One byte can store one character such as 'A' (in ASCII). 4 bits = 1 **nibble**.

2. The storage unit ladder

Unit	Equal to	In bytes	Roughly holds
1 Byte (B)	8 bits	1	One letter
1 Kilobyte (KB)	1024 bytes	2^{10} = 1024	A short text message / half a page
1 Megabyte (MB)	1024 KB	2^{20} = 1,048,576	A 1-minute MP3 song
1 Gigabyte (GB)	1024 MB	2^{30} = 1,073,741,824	About 1 hour of HD video
1 Terabyte (TB)	1024 GB	2^{40} = 1,099,511,627,776	About 250,000 photos

💡 **Rule:** going to a **bigger** unit → **divide by 1024**. Going to a **smaller** unit → **multiply by 1024**. Bits ↔ bytes → × or ÷ **8**.

3. Worked calculations

Question	Working	Answer
2 KB in bytes	2 × 1024	2048 bytes
3 MB in KB	3 × 1024	3072 KB
16 GB in MB	16 × 1024	16,384 MB
2048 GB in TB	2048 ÷ 1024	2 TB
5 bytes in bits	5 × 8	40 bits
4 KB in bits	4 × 1024 × 8	32,768 bits
512 MB in bytes	512 × 1024 × 1024	536,870,912 bytes

Worked example: how many photos fit?

A 64 GB memory card stores photos of 4 MB each. 64 GB = 64 × 1024 MB = 65,536 MB. Photos = 65,536 ÷ 4 = **16,384 photos**.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

💡 **Optional note – KB vs KiB:** in the O/L exam, 1 KB = 1024 bytes. Strictly, the international standard calls 1024 bytes a **kibibyte (KiB)** and uses **kB = 1000 bytes**. That is why a '1 TB' disk shows about 931 GB in Windows.

4. Comparing storage devices

Device	Typical capacity (2026)	Notes
Cache memory	a few MB (e.g. 8–64 MB)	Inside the CPU; fastest, smallest
ROM (firmware)	a few MB (e.g. 16–32 MB)	Holds BIOS/UEFI start-up program
RAM	8 – 16 GB (PC), 4 – 12 GB (phone)	Main memory, volatile
CD / DVD	700 MB / 4.7 GB	Optical, now rarely used
USB flash drive / memory card	32 GB – 1 TB	Portable flash storage
SSD / NVMe	256 GB – 2 TB	Fast flash secondary storage
Hard disk (HDD)	1 – 20 TB	Magnetic, cheap per GB
Cloud storage	15 GB free – many TB	Online, needs internet

Order of capacity (smallest → largest, typical): Cache < ROM < CD < DVD < RAM < USB flash drive < SSD < hard disk. Speed is almost the opposite: cache is the fastest.

NEW 2026 **2026 update:** floppy disks are history. Most new laptops use **NVMe SSDs**, phones use built-in flash (128 GB – 1 TB), and students save work to **cloud storage** such as Google Drive or OneDrive.

Key Words

Byte – A group of 8 bits; stores one character.

Nibble – Half a byte = 4 bits.

Kilobyte (KB) – 1024 bytes (2^{10}).

Megabyte (MB) – 1024 KB = 1,048,576 bytes.

Gigabyte (GB) – 1024 MB; used for RAM and phone storage.

Terabyte (TB) – 1024 GB; used for hard disks and large SSDs.

Storage capacity – The amount of data a device can hold.

Cache memory – Very fast, small memory inside the CPU (a few MB).

SSD – Solid State Drive – fast flash storage with no moving parts.

Memory card – Small flash card (SD/microSD) used in phones and cameras.

Cloud storage – Saving files on internet servers, e.g. Google Drive.

Kibibyte (KiB) – Standard name for exactly 1024 bytes; kB means 1000 bytes.

Sample Q&A

1. How many bits are in one byte?

Answer: 8 bits.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Convert 3 MB to KB.

> **Answer:** 3 × 1024 = 3072 KB.

3. Convert 4096 MB to GB.

> **Answer:** 4096 ÷ 1024 = 4 GB.

4. Arrange cache, hard disk, RAM and DVD from smallest to largest capacity.

> **Answer:** Cache < DVD (4.7 GB) < RAM (8–16 GB) < hard disk.

5. Why does a 1 TB disk show less than 1024 GB on a PC?

> **Answer:** Makers count 1 TB as 1,000,000,000,000 bytes, but the PC divides by 1024, so it shows about 931 GB.

6. Give two advantages of cloud storage.

> **Answer:** Access files from any device anywhere; automatic backup; easy sharing.

MCQ

1. How many bits make one byte?

A. 8
B. 1024
C. 4
D. 16

2. 1 KB is equal to:

A. 1024 bits
B. 1024 bytes
C. 8 bytes
D. 1000 bits

3. How many KB are in 2 MB?

A. 512
B. 2000
C. 2048
D. 1024

4. How many MB are in 16 GB?

A. 16000
B. 8192
C. 16384
D. 1600

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. 2048 GB is equal to:

A. 0.5 TB
B. 20 TB
C. 2 MB
D. 2 TB

6. How many bits are in 4 bytes?

A. 4096
B. 12
C. 32
D. 4

7. Which is the correct order from SMALLEST to LARGEST?

A. KB, bit, byte, GB, MB, TB
B. Byte, bit, KB, MB, TB, GB
C. Bit, KB, byte, MB, GB, TB
D. Bit, byte, KB, MB, GB, TB

8. Which memory usually has the SMALLEST capacity?

A. Cache
B. RAM
C. SSD
D. Hard disk

9. A typical 2026 student laptop has RAM of about:

A. 8 – 16 MB
B. 8 – 16 GB
C. 640 KB
D. 8 – 16 TB

10. A 64 GB memory card stores 4 MB photos. About how many photos fit?

A. 16,384
B. 256
C. 1,600
D. 16

11. Which storage has the LARGEST capacity among these?

A. DVD
B. CD
C. RAM
D. Hard disk (4 TB)

12. A CD can store about:

A. 4.7 TB
B. 7 KB
C. 700 MB
D. 700 GB

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

13. Which storage needs an internet connection to use?

A. Cloud storage
B. Cache
C. SSD
D. USB flash drive

14. In the international standard, 1 kB (small k) means:

A. 1024 bytes
B. 1000 bytes
C. 8 bytes
D. 1000 bits

15. How many bytes are in 1 MB?

A. 1,024
B. 1,048,576
C. 1,000
D. 1,073,741,824

Structured Questions

1. Fill in the blanks: (i) 1 byte = ___ bits (ii) 1 KB = ___ bytes (iii) 1 GB = ___ MB (iv) 1 TB = ___ GB (4 marks) *(4 marks)*

> (i) 8 (ii) 1024 (iii) 1024 (iv) 1024

2. Convert: (i) 5 KB to bytes (ii) 8192 MB to GB (iii) 3 bytes to bits. Show working. (3 marks) *(3 marks)*

> (i) 5 × 1024 = 5120 bytes (ii) 8192 ÷ 1024 = 8 GB (iii) 3 × 8 = 24 bits

3. A student has a 32 GB USB drive. Each video is 512 MB. (i) Convert 32 GB to MB. (1 mark) (ii) How many videos can be stored? (2 marks) *(3 marks)*

> (i) 32 × 1024 = 32,768 MB (ii) 32,768 ÷ 512 = 64 videos

4. Arrange these in ascending order of typical capacity: RAM, cache, SSD, DVD, ROM. (3 marks) *(3 marks)*

> Cache (MB) < ROM (MB) < DVD (4.7 GB) < RAM (8–16 GB) < SSD (256 GB–2 TB).

5. A text file has 2000 characters. Each character uses 1 byte. (i) What is the size in bytes? (1 mark) (ii) Is it more or less than 2 KB? Explain. (2 marks) *(3 marks)*

> (i) 2000 bytes (ii) Less – 2 KB = 2 × 1024 = 2048 bytes, and 2000 < 2048.

6. (i) Give one advantage and one disadvantage of cloud storage. (2 marks) (ii) Why is an SSD preferred over a hard disk in new laptops? (2 marks) *(4 marks)*

> (i) Advantage: access from anywhere / backup / sharing. Disadvantage: needs internet / privacy risk / monthly cost for large space. (ii) Faster start-up and loading, no moving parts so tougher, lighter and uses less power.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Summary

- 1 byte = 8 bits; 1 nibble = 4 bits.
- 1 KB = 1024 bytes; 1 MB = 1024 KB; 1 GB = 1024 MB; 1 TB = 1024 GB.
- Bigger unit → divide by 1024; smaller unit → multiply by 1024.
- Bytes ↔ bits: multiply or divide by 8.
- Capacity: cache < ROM < CD < DVD < RAM < USB < SSD < HDD (typical).
- 2026: RAM 8–16 GB, SSD 256 GB–2 TB, cache a few MB.
- Cloud storage keeps files online and needs the internet.
- kB = 1000 bytes and KiB = 1024 bytes (optional standard).

MCQ answer key – 3.4

1. **A** – 1 byte = 8 bits.
2. **B** – O/L: 1 KB = 1024 bytes.
3. **C** – 2 × 1024 = 2048.
4. **C** – 16 × 1024 = 16384.
5. **D** – 2048 ÷ 1024 = 2.
6. **C** – 4 × 8 = 32.
7. **D** – Each step × 1024 (byte = 8 bits).
8. **A** – Cache is only a few MB.
9. **B** – Modern RAM is measured in GB.
10. **A** – 64 × 1024 ÷ 4 = 16,384.
11. **D** – HDDs reach many TB.
12. **C** – CD ≈ 700 MB; DVD ≈ 4.7 GB.
13. **A** – Cloud files live on internet servers.
14. **B** – kB = 1000 bytes; KiB = 1024 bytes.
15. **B** – 1024 × 1024.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

G.C.E. O/L Grade 10 ICT · Lesson 3

3.5 Coding Systems – BCD, EBCDIC, ASCII and Unicode

When you type the letter **A**, the computer does not store a picture of 'A' – it stores a **code number** in binary. A **coding system** is an agreed table that links characters to binary codes. In this note you will learn **BCD, EBCDIC, ASCII and Unicode**, where each is used, and their limitations.

1. Why do we need coding systems?

Computers understand only 0 and 1. To store letters, digits and symbols, each character is given a unique number. If all computers use the **same code**, a message typed on one device can be read correctly on another.

2. BCD – Binary Coded Decimal

BCD codes **each decimal digit separately in 4 bits**. It is used in digital clocks, calculators and electronic displays where digits are shown one by one.

Decimal	BCD	Pure binary
7	0111	111
59	0101 1001	111011
2026	0010 0000 0010 0110	11111101010

💡 **Note:** BCD of 59 (0101 1001) is NOT the same as binary of 59 (111011). BCD uses only codes 0000–1001; 1010–1111 are not used, so it wastes space and **cannot code letters**.

3. EBCDIC

EBCDIC (Extended Binary Coded Decimal Interchange Code) is an **8-bit** code made by **IBM** for its mainframe computers. It can code 256 characters. Example: 'A' = $C1_{16}$ = 11000001_2.

- Limitation: used mainly on IBM mainframes only – not compatible with ASCII.
- Letters are not in one continuous block, so sorting is harder.

4. ASCII

ASCII (American Standard Code for Information Interchange) is the most common code for English text. **Standard ASCII uses 7 bits → 2^7 = 128 characters** (codes 0–127). **Extended ASCII uses 8 bits → 256 characters**, adding symbols like é, £ and line-drawing characters.

Character	Decimal	Binary (7-bit)	Hex
A	65	1000001	41
B	66	1000010	42
Z	90	1011010	5A
a	97	1100001	61

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

0	48	0110000	30
9	57	0111001	39
space	32	0100000	20

💡 **Exam shortcuts:** 'A' = 65, 'a' = 97 (lower case = upper case + 32), '0' = 48. So 'C' = 67, 'c' = 99 and '5' = 53.

Worked example: the word **CAT** in ASCII = 67 65 84 = 1000011 1000001 1010100.

- Limitation: only 128 (or 256) characters – no space for Sinhala, Tamil, Chinese or emoji.

5. Unicode

Unicode gives a unique code (code point) to **every character in every language**, plus symbols and emoji – over 150,000 characters. It is written as U+ followed by hex. The first 128 Unicode codes are the same as ASCII. It is stored using **UTF-8** (1–4 bytes per character) or UTF-16.

Character	Code point	UTF-8 bytes
A	U+0041	1
අ	U+0D85	3
ක	U+0D9A	3
😀	U+1F600	4

NEW 2026 **Sinhala in Unicode:** Sinhala letters live in the block **U+0D80 – U+0DFF** (e.g. අ = U+0D85). Thanks to Unicode you can type Sinhala on Facebook, WhatsApp and this website, and emoji like 😀 (U+1F600) work on every phone.

- Limitation: needs more storage than ASCII (2–4 bytes for many characters).
- Old software and devices may not show all characters (boxes □ appear).

6. Comparison

Code	Bits	Characters	Main use	Limitation
BCD	4 per digit	10 digits	Clocks, calculators	Digits only
EBCDIC	8	256	IBM mainframes	Not compatible with ASCII
ASCII	7 (8 extended)	128 (256)	English text, keyboards	No other scripts
Unicode	8–32 (UTF-8: 1–4 bytes)	150,000+	All languages, emoji, web	Uses more space

Key Words

Coding system – An agreed table that gives each character a binary code.

Character – A letter, digit, symbol or space.

BCD – Binary Coded Decimal – each decimal digit coded in 4 bits.

EBCDIC – IBM's 8-bit character code for mainframes.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

ASCII – 7-bit code for 128 characters; 'A' = 65.	**Extended ASCII** – 8-bit version of ASCII with 256 characters.
Unicode – Universal code covering all languages and emoji.	**Code point** – The number Unicode gives a character, e.g. U+0D85.
UTF-8 – A way of storing Unicode using 1–4 bytes per character.	**Sinhala Unicode block** – The range U+0D80–U+0DFF used for Sinhala letters.
Emoji – Picture characters such as 😀, coded in Unicode.	**Compatibility** – Ability of different systems to read the same data correctly.

Sample Q&A

1. What is a coding system?

> **Answer:** An agreed set of binary codes for characters, so all computers store and read text the same way.

2. Write 59 in BCD.

> **Answer:** 5 → 0101, 9 → 1001 → 0101 1001.

3. How many characters can 7-bit ASCII represent? Why?

> **Answer:** 128, because $2^7 = 128$.

4. The ASCII code of 'A' is 65. What is the code of 'E' and of 'e'?

> **Answer:** 'E' = 69; 'e' = 101 (add 32).

5. Why was Unicode introduced?

> **Answer:** ASCII could not represent languages such as Sinhala, Tamil or Chinese. Unicode gives a code to every character in every language.

6. Which Unicode range is used for Sinhala?

> **Answer:** U+0D80 to U+0DFF.

MCQ

1. How many characters can standard 7-bit ASCII represent?

A. 65,536
B. 256
C. 7
D. 128

2. The ASCII code of 'A' is 65. What is the code of 'D'?

A. 100
B. 67
C. 68
D. 69

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. The ASCII code of lower-case 'a' is:

A. 97
B. 1
C. 32
D. 65

4. Which coding system was developed by IBM for mainframes?

A. Unicode
B. EBCDIC
C. ASCII
D. BCD

5. What is 47 in BCD?

A. 0100 0111
B. 0100 0110
C. 0111 0100
D. 101111

6. Which coding system can represent Sinhala letters?

A. BCD
B. EBCDIC
C. Unicode
D. 7-bit ASCII

7. How many characters can extended ASCII represent?

A. 512
B. 128
C. 256
D. 1024

8. BCD is mainly used in:

A. Web page Sinhala text
B. Emoji on phones
C. Image compression
D. Digital clocks and calculators

9. A main limitation of ASCII is that it:

A. Uses 32 bits per character
B. Cannot represent the letter A
C. Cannot represent most non-English scripts
D. Works only on IBM mainframes

10. The Unicode code point of the Sinhala letter අ is:

A. U+0D85
B. U+1F600
C. U+0D00
D. U+0041

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

11. The ASCII code of the digit '0' is:

A. 65
B. 48
C. 0
D. 30

12. The word 'HI' is stored in ASCII as:

A. 71 72
B. 72 73
C. 104 105
D. 8 9

13. Emoji such as 😀 can be stored because of:

A. Unicode
B. EBCDIC
C. BCD
D. 7-bit ASCII

14. In UTF-8, how many bytes does the Sinhala letter ඇ use?

A. 1
B. 4
C. 2
D. 3

15. The 7-bit ASCII code 1000010 represents:

A. 2
B. B
C. b
D. A

Structured Questions

1. (i) What does ASCII stand for? (1 mark)
(ii) How many bits does standard ASCII use and how many characters can it code? (2 marks)
(iii) State one limitation of ASCII. (1 mark) *(4 marks)*

> (i) American Standard Code for Information Interchange (ii) 7 bits; $2^7 = 128$ characters (iii) Cannot code Sinhala/Tamil/Chinese or emoji.

2. The ASCII code of 'A' is 65. (i) Write the ASCII code of 'G' in decimal. (1 mark) (ii) Convert it to 7-bit binary. (2 marks) (iii) What is the code of 'g'? (1 mark) *(4 marks)*

> (i) 71 (ii) 1000111_2 (iii) 103

3. (i) Write 2026 in BCD. (2 marks) (ii) Write 2026 in pure binary. (2 marks) (iii) Which uses more bits? (1 mark) *(5 marks)*

> (i) 0010 0000 0010 0110 (ii) 11111101010_2 (iii) BCD uses 16 bits; binary uses 11 bits, so BCD uses more.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

4. Compare ASCII and Unicode under: (i) number of characters (ii) languages supported (iii) storage needed. (3 marks) *(3 marks)*

> (i) ASCII 128/256; Unicode 150,000+. (ii) ASCII – English only; Unicode – all languages incl. Sinhala, Tamil, plus emoji. (iii) ASCII 1 byte; Unicode (UTF-8) 1–4 bytes.

5. A message is stored in 7-bit ASCII as: 1000011 1000001 1010100. (i) Convert each code to decimal. (2 marks) (ii) Decode the word. (1 mark) *(3 marks)*

> (i) 67, 65, 84 (ii) CAT

6. A student types a Sinhala message on WhatsApp with a 😀 emoji. (i) Which coding system makes this possible? (1 mark) (ii) Give the Unicode range for Sinhala. (1 mark) (iii) Give one reason EBCDIC is not used for this. (1 mark) *(3 marks)*

> (i) Unicode (UTF-8) (ii) U+0D80 – U+0DFF (iii) EBCDIC has only 256 codes and is used mainly on IBM mainframes – no Sinhala or emoji.

Summary

- A coding system gives every character a binary code.
- BCD codes each decimal digit in 4 bits (clocks, calculators).
- EBCDIC: 8-bit IBM mainframe code, 256 characters.
- ASCII: 7 bits = 128 characters; extended ASCII 8 bits = 256.
- ASCII: 'A' = 65, 'a' = 97, '0' = 48.
- Unicode covers all languages and emoji; written as U+hex.
- Sinhala Unicode block: U+0D80 – U+0DFF; stored with UTF-8.
- Limits: BCD digits only, EBCDIC IBM only, ASCII English only, Unicode uses more space.

MCQ answer key – 3.5

1. **D** – $2^7 = 128$.
2. **C** – A65 B66 C67 D68.
3. **A** – 'a' = 'A' + 32 = 97.
4. **B** – EBCDIC = IBM, 8-bit.
5. **A** – 4 → 0100, 7 → 0111.
6. **C** – Sinhala block U+0D80–U+0DFF.
7. **C** – 8 bits → $2^8 = 256$.
8. **D** – BCD shows decimal digits one by one.
9. **C** – Only 128/256 characters.
10. **A** – Sinhala block starts at U+0D80; අ = U+0D85.
11. **B** – '0' = 48 (30 in hex).
12. **B** – H = 72, I = 73.
13. **A** – 😀 = U+1F600.
14. **D** – Characters from U+0800 to U+FFFF use 3 bytes.
15. **B** – $1000010_2 = 66$ = 'B'.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.